AF/2182

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application No.: 09/828,494	§	Examiner: Casiano, Angel L.
Filed: April 5, 2001	§	Group/Art Unit: 2182
Inventor(s):	§	Atty. Dkt. No: 5181-86600
Brown et al.	§	
	§	
	§	I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below.
	§	
	§	Stephen J. Curran
	§	Printed Name
	§	Signature 11/23/04 Date
Title: Configuring System Units Using On-board Class Information	§	
	§	
	§	
	§	
	§	
	§	
	§	

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal of October 1, 2004, Appellants present this Appeal Brief. Appellants respectfully request that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The subject application is owned by Sun Microsystems Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at , as evidenced by the assignment recorded at Reel 012096, Frame 0527.

II. RELATED APPEALS AND INTERFERENCES

No other appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1, 4-6, 9-13, 18-31, and 34 are pending in the present application. Claims 1, 4-6, 9-13, 18-31, and 34 stand finally rejected and are the subject of this appeal. A clean copy of claims 1, 4-6, 9-13, 18-31, and 34, as on appeal (incorporating all amendments), is included in the Appendix hereto.

IV. STATUS OF AMENDMEMNTS

No amendment to the claims has been filed subsequent to the final rejection. The Appendix hereto reflects the current state of the rejected claims.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The invention relates to the integration of a unit in an apparatus such as, for example, configuring a field replaceable unit (FRU) into apparatus such as a computer system. FRUs can be used in many different systems. They find particular but not exclusive application in computer systems; particularly fault tolerant computer systems where it may be desirable to be able to readily replace units which have developed a fault or have been superseded by a more recent version. Examples of FRUs for such a system can

include, for example, a CPU, a PCI card, power supply units (PSUs), a motherboard, or any other system components. One FRU, for example a field replaceable card, can include hardware for implementing several devices (e.g. a multiple Ethernet adapter, or a SCSI adapter with an Ethernet adapter). Some conventional FRUs include non-volatile memory (e.g. EEPROMs), which can contain information relating to the FRU. In some systems, FRUs can include basic FRU identification information in the non-volatile memory. (*See Specification*, Page 1)

In addition, some conventional systems include a system management suite, collectively known as a configuration management system (CMS). A CMS may manage the FRUs, and other devices and system resources using objects to represent the FRUs. An object forms a particular instance of a CMS class, which is defined by a CMS definition (CMSDEF). (*See* Page 1)

For example, a CAF (Console and Fans unit) CMSDEF defines the CAF CMS class of which the object CAF_1 is an instance that represents a particular CAF FRU. The CAF_1 object may have an attribute called LOCATION having the value A_CAF, indicating that the FRU represented by the CAF_1 object has been inserted into location A_CAF in the chassis of the computer system. (*See Specification*, Page 1)

However, it may be difficult when initiating a system to establish an initial configuration for the system by supplying initial values to object attributes which represent that configuration. In some conventional systems, the CMS may use a chassis type number read from the EEPROM of a control-panel FRU to establish a default configuration for the system. However, this may provide only a crude configuration for the system, as it relied in effect on 'fine-tuning' a pre-defined configuration to the specific needs of that type of system. Only 'standard' parts of the configuration (e.g. the boot disks and their controllers and the CPUs) could be established in this way so much of the more complex configuration (e.g. serial ports) had to be performed manually. (*See Specification*, Page 1, 2)

In accordance with one aspect of the invention, there is provided a method of automatic configuration of a unit (e.g. a field replaceable unit (FRU)) of an apparatus. The method includes accessing class information held in the unit upon insertion of the unit into the apparatus and prior to integrating the unit functionally in the apparatus. The class information may represent an object class for the unit. The method also includes using the accessed class information to reference, in storage in the apparatus separate from the unit, object definitions for the class of unit. The object definitions include initialization code operable on receipt of the accessed class information to produce configuration information operable to produce object configuration statements for the unit. The object configuration statements include at least one of the following: the object class for the unit; an object instance number; an attribute name; and a value for the attribute. The method also includes verifying the validity of the configuration information and, when the configuration information is valid, storing the configuration information in a configuration file for the apparatus. (*See Specification*, Page 3, claims)

In accordance with another aspect of the invention, an apparatus includes a plurality of units that include unit storage for holding class information for the unit and a configuration mechanism. The configuration mechanism may be operable to access class information held in the unit upon insertion of the unit into the apparatus and prior to integrating the unit functionally in the apparatus. The class information may represent an object class for the unit. The configuration mechanism may also be operable includes use the accessed class information to reference, in storage in the apparatus separate from the unit, object definitions for the class of unit. The object definitions include initialization code operable on receipt of the accessed class information to produce configuration information operable to produce object configuration statements for the unit. The object configuration statements include at least one of the following: the object class for the unit; an object instance number; an attribute name; and a value for the attribute. The configuration mechanism may be operable to verify the validity of the configuration

information and, when the configuration information is valid, store the configuration information in a configuration file for the apparatus. (*See Specification*, Page 3, claims)

The class information can be held in non-volatile memory (e.g., in an EEPROM) in the unit. This information can be read on inserting the unit into the system and can be used to establish the initial configuration prior to full integration of the unit into the system. (*See Specification*, Page 4)

In an embodiment of the invention, a unit contains information defining one or more configuration management system (CMS) classes for a FRU. A management class is identified for managing the FRU. (*See Specification*, Page 4)

Verification of the derived configuration information can be employed to check on the operability and compatibility of the unit to other units in the system prior to integration thereof. (*See Specification*, Page 4)

More detailed information regarding the unit, for example relating to the configuration of devices in the unit, can be effected in a second stage. For example, in an embodiment of the invention, an FRU contains information defining a configuration management system (CMS) class for the unit. The unit can include one or more devices (resources), and each device can be associated with its own CMS class as well. The CMS class information for the unit can be accessed and used to derive the initial configuration information for the unit. The class information for the devices can then be accessed and used for further configuring those devices. (*See Specification*, Page 4)

The CMS class information stored in the unit can be in the form of a name for the class of unit, which is used to identify or point to configuration code for configuring that class of unit. The configuration code can form part of the object definitions (CMS definitions) which are held outside the units, for example in computer system memory, on a disk, or at a remote site via a telecommunication interface. The CMS class information

effectively performs the function of providing a handle for accessing the means for generating the initial configuration. (*See Specification*, Pages 4, 5)

The operations indicated above are effected on initiation of the system. However, they could optionally be also effected during running of the system for changing a configuration. (*See Specification*, Page 7)

Although the invention finds particular application to a configuration management system responsive to configuration management system definitions, the invention could also be applied to other forms of system and network management. For example, in a Telecommunications Management Network (TMN) environment, the memory of a unit could contain (either directly, or via a reference to a disk file) the GDMO definitions of the unit and its devices, and these could be passed to a local agent and a remote manager to allow the unit to be managed. (*See Specification*, Pages 7, 8)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APEAL

1. Claims 1, 4-5, 9-13, 20-31, and 34 are rejected under 35 U.S.C. § 103(a) as being patentable over Sudhakaran et al. (U.S. Patent Application Number 2003/0014468) in view of Cepulis (U.S. Patent Number 6,397,268).

2. Claims 6, 18, and 19 are rejected under 35 U.S.C. § 103(a) as being patentable over Sudhakaran et al. in view of Macon Jr., et al. (U.S. Patent Number 5,752,249).

VII. ARGUMENT

A. Claims 1, 4-6, 9-13, 18-31, and 34

The Examiner rejected claims 1, 4-5, 9-13, 20-31, and 34 as being obvious over Sudhakaran et al. in view of Cepulis under 35 U.S.C. § 103(a). Appellants respectfully traverse this rejection in light of the following remarks.

The Examiner rejected claims 6, 18, and 19 as being obvious over Sudhakaran et al. in view of Macon Jr., et al. under 35 U.S.C. § 103(a). Appellants respectfully traverse this rejection in light of the following remarks.

The Examiner has acknowledged that Sudhakaran **does not teach** "object definitions which include initialization code operable on receipt of the accessed class information to produce configuration information operable to produce object configuration statements for the unit, that comprise at least one of the following: the object class for the unit; an object instance number; an attribute name; and a value for the attribute..." The Examiner also acknowledged that Sudhakaran **does not teach** verifying "the validity of the configuration information and, when the configuration information is valid, store the configuration information in a configuration file for the apparatus including a location of the unit in the apparatus to enable functional integration of the unit in the apparatus."

The Examiner has however, asserted that Cepulis teaches these features. The Appellant respectfully disagrees with the Examiner's assertions and his characterization of Cepulis.

Specifically, Cepulis is directed toward tracking **PCI** bus numbers that change during re-configuration, wherein at col. 2, lines 48-64 Cepulis discloses

> "However, situations arise more and more often which require rerunning the system configuration utility to update the device configuration information stored in the NVRAM when a new device is added to the computer system. One specific situation is when a PCI peripheral device interface card having a PCI-PCI bridge is placed into a PCI connector slot of a first PCI bus of the computer system. The PCI-PCI bridge, which creates a new PCI bus, causes the PCI bus numbers of all subsequent PCI buses to

> increase by one (PCI-PCI bridge may be a PCI interface card having its own PCI bus for a plurality of PCI devices integrated on the card or for PCI bus connector slots associated with the new PCI bus). This creates a problem since any user configured information such as interrupt request (IRQ) number, controller order number, etc., stored in the NVRAM specifies the bus and device/function number of the PCI device to which it applies." (Emphasis added)

Cepulis also discloses at col. 4, lines 21-34

> "Each PCI card comprises at least one PCI device that is unique in the computer system. Each PCI device has a plurality of registers containing unique criteria such as Vender ID, Device ID, Revision ID, **Class Code Header** Type, etc. Other registers within each PCI device may be read from and written to so as to further coordinate operation of the PCI devices in the computer system. During system configuration,, each PCI device is discovered and its personality information such as interrupt request number, bus master priority, latency time and the like are stored in the system non-volatile random access memory (NVRAM) using, for example, the ESCD format." (Emphasis added)

Further Cepulis discloses at col. 6, lines 8-15

> "In the present invention, the PCI bus number, PCI device number and PCI physical slot number are found in both the ESCD freeform information structure ECD_PCIBRDID and IRQ routing table. During computer system startup, the PCI bus number in the IRQ routing table is updated for each PCI device. The ESCD freeform information structure ECD_PCIBRDID stored in the NVRAM, typically, is not updated during system startup." (Emphasis added)

From the foregoing, it appears that Cepulis teaches detecting when a PCI device bus number has changed and updating the storages accessed by auto-configuration BIOS. However, Cepulis **does not**, as the Examiner has suggested, teach or suggest "using the **accessed class information** to reference ... object definitions for the class of unit, which object definitions include initialization code operable on receipt of the accessed class information to produce configuration information operable to produce object configuration statements for the unit, that comprise at least one of the following: the object class for the unit; an object instance number; an attribute name; and a value for the attribute;" as recited in Appellant's claim 1 (Emphasis added)

The Class Code Header Type referenced by Cepulis is a required field in the PCI header which identifies the generic function of the device. However, Cepulis is silent on the use of this Class information and after col. 4, never mentions class again. More particularly, Cepulis uses information such as the PCI bus number, PCI device number and PCI physical slot number, **not** the class information.

In addition, Cepulis is completely silent on verifying "the validity of the configuration information." Accordingly, Cepulis **does not teach or suggest** "the validity of the configuration information and, when the configuration information is valid, store the configuration information in a configuration file for the apparatus including a location of the unit in the apparatus to enable functional integration of the unit in the apparatus" as recited in Appellant's claim 1.

Accordingly, even if (*arguendo*) one were to combine the references as the Examiner has suggested, Appellant respectfully submits one would not arrive at the subject matter recited in claim 1.

Furthermore, the above notwithstanding, the Appellant asserts that there is no motivation to combine the references in the manner in which the Examiner has suggested. According to MPEP §2143.01, "The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art suggests the desirability of the combination. *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990) ..." Thus, in contrast to the Examiner, Appellant cannot find any such suggestion or motivation.

More particularly, the Examiner cites the suggestion to combine the references as "a more robust and complete plug and play implementation of the computer system" being found at col. 6, lines 49-51 of Cepulis. The Appellant respectfully asserts that this quote is taken out of context by the Examiner. Cepulis, in fact, discloses at col. 6, lines 41-51 "Another feature of the present invention is that **any bus number changes** found

during POST of the contents of the IRQ routing table and ECD_PCIBRDID **may be corrected by overwriting the RAM locations containing the incorrect bus numbers** for the affected PCI devices. The computer system can then utilize the corrected bus numbers in the system RAM to perform the necessary steps for startup and proper operation of the computer system. **This feature makes for** a more robust and complete plug and play implementation of the computer system."

The above feature(s) disclosed in Cepulis, which are alleged to produce the suggested improvements, are not only not recited in Appellant's claim 1, they are not the topic of the Appellant's disclosure. Accordingly, Appellant submits that the Examiner's asserted suggestion for motivation to combine the references is erroneous.

Macon, Jr. (first paragraph of summary) describes "a way of allowing one or more instantiated parameterized collection classes to survive the termination or loss of a routine that instantiated them". Appellant submits that Macon, Jr. **does not teach or suggest** "object definitions include initialization code operable on receipt of the accessed class information to produce configuration information operable to produce object configuration statements for the unit, that comprise at least one of the following: the object class for the unit; an object instance number; an attribute name; and a value for the attribute."

Thus, neither Sudhakaran, Cepulis, nor Macon, Jr., taken either singly or in combination, teach or suggest the combination of features recited in Appellant's claim 1.

For the foregoing reasons, Appellant submits that claim 1, along with its dependent claims, patentably distinguishes over Sudhakaran in view of Cepulis, and over Sudhakaran in view of Cepulis, and in further view of Macon, Jr.

Claims 13, 28 and 31 recite features that are similar to the features recited in

Appellant's claim 1. Accordingly, Appellant believes that claims 13, 28 and 31, along with their respective dependent claims, patentably distinguish over Sudhakaran in view of Cepulis, and over Sudhakaran in view of Cepulis, and in further view of Macon, Jr. for at least the reasons given above.

VII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1, 4-6, 9-13, 18-31, and 34 was erroneous, and reversal of his decision is respectfully requested.

Respectfully submitted,

Stephen J. Curran
Reg. No. 50,664
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: November 23, 2004

IX. APPENDIX

The claims on appeal are as follows.

1. A method of automatic configuration of a unit forming a component of an apparatus, the method comprising:

a) accessing class information held in the unit on insertion of the unit into the apparatus prior to integrating the unit functionally in the apparatus, said class information representing an object class for the unit;

b) using the accessed class information to reference, in storage in the apparatus separate from the unit, object definitions for the class of unit, which object definitions include initialization code operable on receipt of the accessed class information to produce configuration information operable to produce object configuration statements for the unit, that comprise at least one of the following: the object class for the unit; an object instance number; an attribute name; and a value for the attribute; and

c) verifying the validity of the configuration information and, when the configuration information is valid, storing the configuration information in a configuration file for the apparatus including a location of the unit in the apparatus to enable functional integration of the unit in the apparatus.

4. The method of claim 1, further comprising accessing the unit when the unit is functionally integrated in the apparatus for further configuration data held therein.

5. The method of claim 4, wherein the further configuration data comprises a device object class and device object attributes for a device of the unit.

6. The method of claim 1, wherein the class information is held in non-volatile memory in the unit.

9. The method of claim 1 for configuring a plurality of units for a configuration management system, wherein the class information identifies at least one configuration management system class for the unit.

10. The method of claim 1, said apparatus having a plurality of locations for receiving a said unit and wherein the method comprises probing each said location in the apparatus for class information held in a unit at that location.

11. The method of claim 10, wherein, in step (c), a set of object configuration statements for respective units are stored in the configuration file.

12. The method of claim 1, wherein the unit is a field replaceable unit.

13. Apparatus comprising:

a plurality of units that each include unit storage for holding class information for the unit that represents an object class for the unit; and

a configuration mechanism operable to:

a) access class information held in the unit on insertion of the unit into the apparatus prior to integrating the unit functionally in the apparatus, said class information representing an object class for the unit;

b) use the accessed class information to reference, in storage in the apparatus separate from the unit, object definitions for the class of unit, which object definitions include initialization code operable on receipt of the accessed class information to produce object configuration statements for the unit, that comprise at least one of the following: the object class for the unit; an object instance number; an attribute name; and a value for the attribute; and

c) verify the validity of the configuration information and, when the configuration information is valid, store the configuration information in a configuration file for the apparatus including a location of the unit in the apparatus to enable functional integration of the unit in the apparatus.

18. The apparatus of claim 13, wherein the unit storage comprises non-volatile memory.

19. The apparatus of claim 18, wherein the non-volatile memory is an EEPROM.

20. The apparatus of claim 13, wherein the configuration mechanism is responsive to derived configuration information to verify the validity of the configuration information prior to storage thereof in the system storage.

21. The apparatus of claim 13, wherein the configuration mechanism is part of a configuration management system and the class information identifies at least one configuration management system class for the unit.

22. The apparatus of claim 21, comprising a chassis for a plurality of units locatable within the chassis.

23. The apparatus of claim 22, wherein the configuration mechanism probes each location in the apparatus for receiving a said unit for accessing class information held in a said unit at that location.

24. The apparatus of claim 23, comprising a configuration file in system storage for persistent storage of a set of object configuration statements for respective units.

25. The apparatus of claim 13, wherein a said unit is a field replaceable unit.

26. The apparatus of claim 13 forming a computer system.

27. The apparatus of claim 26, wherein the computer system is a fault-tolerant computer system.

28. A configuration management system operable on apparatus that includes a plurality of units that each have unit storage for holding class information that represents an object class for the unit, the configuration management system comprising an initialization component configured to:

a) access class information held in the unit on insertion of the unit into the apparatus prior to integrating the unit functionally in the apparatus, said class information representing an object class for the unit;

b) use the accessed class information to reference, in storage in the apparatus separate from the unit, object definitions for the class of unit, which object definitions include initialization code operable on receipt of the accessed class information to produce object configuration statements for the unit, that comprise at least one of the following: the object class for the unit; an object instance number; an attribute name; and a value for the attribute; and

c) verify the validity of the configuration information and, when the configuration information is valid, store the configuration information in a configuration file for the apparatus including a location of the unit in the apparatus to enable functional integration of the unit in the apparatus.

29. The configuration management system of claim 28, wherein the initialization component is configured to probe each location in the apparatus for receiving a unit and, when a location is occupied by a unit, to read class information from storage in the unit.

30. The configuration management system of claim 28, wherein the initialization component is configured to access class information from a unit on insertion of the unit

into the apparatus and to generate object definitions for the unit prior to functional integration of the unit.

31. A carrier medium carrying program means embodying a configuration management operable on apparatus that includes a plurality of units that each have unit storage for holding class information that represents an object class for the unit, the configuration management system comprising an initialization component configured to:

a) access class information held in the unit on insertion of the unit into the apparatus prior to integrating the unit functionally in the apparatus, said class information representing an object class for the unit;

b) use the accessed class information to reference, in storage in the apparatus separate from the unit, object definitions for the class of unit, which object definitions include initialization code operable on receipt of the accessed class information to produce object configuration statements for the unit, that comprise at least one of the following: the object class for the unit; an object instance number; an attribute name; and a value for the attribute; and

c) verify the validity of the configuration information and, when the configuration information is valid, store the configuration information in a configuration file for the apparatus including a location of the unit in the apparatus to enable functional integration of the unit in the apparatus.

34. The carrier medium of claim 31, wherein the storage is operable to record status information relating to system operation for providing an operating history for the unit.

X. EVIDENCE APPENDIX

No evidence submitted under 37 C.F.R. §§ 1.130, 1.131, or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application No.: 09/828,494
Filed: April 5, 2001
Inventor(s):
Brown et al.

Title: Configuring System Units Using On-board Class Information

§ § § § § § § § § § § § § §

Examiner: Casiano, Angel L.
Group/Art Unit: 2182
Atty. Dkt. No: 5181-86600

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below.

Stephen J. Curran
Printed Name


Signature

November 23, 2004
Date

FEE AUTHORIZATION

Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

The Commissioner is hereby authorized to charge the following fee to Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C. Deposit Account Number 501505/5181-86600/SJC:

Fee:	Appeal Brief
Amount:	$340.00
Attorney Docket No.:	5181-86600

The Commissioner is also authorized to charge any extension fee or other fees, which may be necessary to the same account number.

Respectfully submitted,

Stephen J. Curran
Reg. No. 50,664
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: November 23, 2004